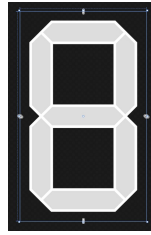




LİSER CONTROL

Display (Class)

User Control



```

1  <UserControl x:Class="DijitalSaat.Display"
2  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
3  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
4  xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
5  xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
6  xmlns:local="clr-namespace:DijitalSaat"
7  mc:Ignorable="d"
8  d:DesignHeight="300" d:DesignWidth="180">
9
10 <Grid>
11 <Viewbox Stretch="Fill">
12 <Image x:Name="imgDisplay" Source="Resimler/bos.png" Height="300" Width="180"/>
13 </Viewbox>
14 </Grid>
15 </UserControl>

```

```

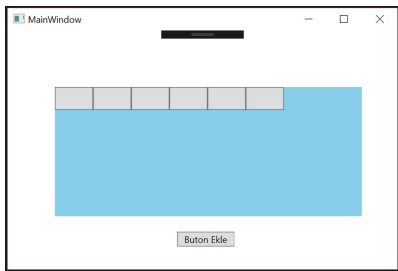
15 namespace DijitalSaat
16 {
17     /// <summary>
18     /// Interaction logic for Display.xaml
19     /// </summary>
20     public partial class Display : UserControl
21     {
22         List<BitmapImage> Resimler;
23         public Display() //yapıcı method
24         {
25             InitializeComponent();
26             ResimlerDoldur();
27         }
28         private void ResimlerDoldur()
29         {
30             Resimler = new List<BitmapImage>();
31             string yol = "Resimler/";
32             Resimler.Add(new BitmapImage(new Uri(yol + "0.png", UriKind.Relative)));
33             Resimler.Add(new BitmapImage(new Uri(yol + "1.png", UriKind.Relative)));
34             Resimler.Add(new BitmapImage(new Uri(yol + "2.png", UriKind.Relative)));
35             Resimler.Add(new BitmapImage(new Uri(yol + "3.png", UriKind.Relative)));
36             Resimler.Add(new BitmapImage(new Uri(yol + "4.png", UriKind.Relative)));
37             Resimler.Add(new BitmapImage(new Uri(yol + "5.png", UriKind.Relative)));
38             Resimler.Add(new BitmapImage(new Uri(yol + "6.png", UriKind.Relative)));
39             Resimler.Add(new BitmapImage(new Uri(yol + "7.png", UriKind.Relative)));
40             Resimler.Add(new BitmapImage(new Uri(yol + "8.png", UriKind.Relative)));
41             Resimler.Add(new BitmapImage(new Uri(yol + "9.png", UriKind.Relative)));
42             Resimler.Add(new BitmapImage(new Uri(yol + "bos.png", UriKind.Relative)));
43         }
44     }
45 }
46
47 private int rakam;
48
49 public int Rakam
50 {
51     get { return rakam; }
52     set
53     {
54         if (value >= 0 && value <= 9) // value 0'la 9 arasındaysa
55         {
56             rakam = value;
57             imgDisplay.Source = Resimler[rakam];
58         }
59         //listemizdeki bir value değeri gelmesi durumunda Resimler dizisinden olmayan bir elemanın okunmasına engel oluyoruz.
60         rakam = -1; //hatalı @:is:
61         imgDisplay.Source = Resimler[0];
62     }
63 }
64
65
66
67
68
69
70

```

```

private void Button_Click(object sender, RoutedEventArgs e)
{
    Button b = new Button();
    b.Height = 30;
    b.Width = 50;
    bizimPanel.Children.Add(b);
}

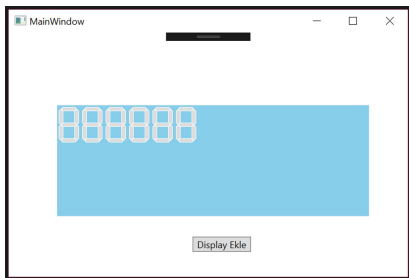
```



```

private void Button_Click(object sender, RoutedEventArgs e)
{
    Display d = new Display();
    d.Height = 50;
    d.Width = 30;
    bizimPanel.Children.Add(d);
}

```



1.Aşama

Display nesnesi oluşur oluşmaz List yapısındaki dinamik bir diziye, kullanacağımız tüm resimlerin doldurulmasını sağlayalım.

```
List<BitmapImage> Resimler;

private void ResimlerDoldur()
{
    Resimler = new List<BitmapImage>();
    string yol = "Resimler/";
    Resimler.Add(new BitmapImage(new Uri(yol + "0.png", UriKind.Relative)));
    Resimler.Add(new BitmapImage(new Uri(yol + "1.png", UriKind.Relative)));
    Resimler.Add(new BitmapImage(new Uri(yol + "2.png", UriKind.Relative)));
    Resimler.Add(new BitmapImage(new Uri(yol + "3.png", UriKind.Relative)));
    Resimler.Add(new BitmapImage(new Uri(yol + "4.png", UriKind.Relative)));
    Resimler.Add(new BitmapImage(new Uri(yol + "5.png", UriKind.Relative)));
    Resimler.Add(new BitmapImage(new Uri(yol + "6.png", UriKind.Relative)));
    Resimler.Add(new BitmapImage(new Uri(yol + "7.png", UriKind.Relative)));
    Resimler.Add(new BitmapImage(new Uri(yol + "8.png", UriKind.Relative)));
    Resimler.Add(new BitmapImage(new Uri(yol + "9.png", UriKind.Relative)));
    Resimler.Add(new BitmapImage(new Uri(yol + "bos.png", UriKind.Relative)));
}

}
```

```
namespace DijitalSaat
{
    /// <summary>
    /// Interaction logic for Display.xaml
    /// </summary>
    public partial class Display : UserControl
    {
        public Display()
        {
            InitializeComponent();
        }
    }
}
```



2.Aşama

Display nesnesine "Rakam" özelliği (property) ekleyelim (NYP'nin hangi avantajı?)

```
private int rakam;

public int Rakam
{
    get { return rakam; }
    set
    {
        if (value >= 0 && value <= 9) // value 0'le 9 arasında
        {
            rakam = value;
            imgDisplay.Source = Resimler[rakam];
        }
        else // istemediğimiz bir value değeri gelmesi durumunda Resimler dizisinden olmayan bir elemanın okunmasını engelliyoruz.
        {
            rakam = -1; // hatalı giriş
            imgDisplay.Source = Resimler[10];
        }
    }
}
```

```
private void Button_Click(object sender, RoutedEventArgs e)
{
    Button b = new Button();
    b.Height = 30;
    b.Width = 50;
    b.Content = "Merhaba";
    bizimPanel.Children.Add(b);
}
```



```
private void Button_Click(object sender, RoutedEventArgs e)
{
    Display d = new Display();
    d.Height = 50;
    d.Width = 30;
    d.Rakam = 5;
    bizimPanel.Children.Add(d);
}
```

